

1 METHOD AND SYSTEM FOR FILE BLOCKING IN AN
2 ELECTRONIC MESSAGING SYSTEM
3

4 This application claims the benefit of U.S. Provisional Application No.
5 60/212,679 titled "METHOD AND SYSTEM FOR FILE BLOCKING IN AN
6 ELECTRONIC MESSAGING SYSTEM", filed June 20, 2000, which is herein
7 incorporated by reference in its entirety.
8

9 BACKGROUND OF THE INVENTION

10 Field of the Invention

11 The present invention relates to the field of computer security and integrity within
12 an electronic messaging environment, and in particular, to a system and method for
13 controlling damage caused by and for containing the spread of software viruses, worms,
14 or other destructive applications in an electronic messaging system.
15

16 Description of the Related Art

17 Electronic messaging began in the early 1970s during the development of the
18 Advanced Research Projects Agency Network (ARPANET), a computer networking
19 project funded by the United States government, designed to share computing, and the
20 predecessor of today's Internet. At the outset, electronic messaging was not even
21 contemplated as a potential application for ARPANET, though Ray Tomlinson's e-mail
22 system would soon become the biggest use of ARPANET.

23 Early e-mail systems provided little more than a mechanism for exchanging text
24 messages. If a user wished to exchange binary files such as executable applications, he
25 or she would use the File Transfer Protocol (FTP). As the use of e-mail grew, network
26 users began to develop ways to send files as attachments; however, e-mail systems were
27 not well suited for transporting binary data because some binary files contained

1 characters or control sequences that confused the e-mail transport and delivery systems.
2 This prompted the development of programs such as "uuencode" that could convert
3 binary files into text files that could then be sent within e-mails.

4 Over the years, e-mail systems have evolved to become more attuned to the
5 demands of users, making common tasks easier to accomplish and creating complex
6 systems that can be customized for the particular demands of an individual or company.
7 For example, one of the most common tasks a user encounters is entering the recipients
8 (e.g., destination addresses) of e-mail messages. The present version of Microsoft
9 Outlook eases the task by including a directory of e-mail users and includes a Visual
10 Basic extension that permits developers to create customized e-mail applications

11 Along with the flexibility of modern e-mail systems comes added risks. Using
12 modern e-mail clients such as Microsoft Outlook, users can send applications as
13 attachments to e-mail messages. By making it easy for users to execute attached
14 applications, modern electronic messaging systems prove easy targets for malicious
15 applications such as electronic viruses, worms, and other destructive programs. The
16 terms "virus" and "worm" are used herein to generically refer to those malicious
17 applications and destructive programs.

18 A recent outbreak that took the world by storm was the "I Love You" worm.
19 This Visual Basic code, when executed, looks through a user's address book and sends
20 an e-mail message with a copy of the worm to each user in that address book. The worm
21 spread rapidly, clogging e-mail servers throughout the world and forcing large
22 corporations to shut down e-mail servers for an entire day. Network Associates, a major
23 provider of anti-virus, network security and management software, estimates that the
24 worm caused approximately \$6.7 billion in damages throughout the world.

25 Once a virus has been identified, there are many anti-virus applications that can
26 help enterprises detect and prevent outbreaks; however, when a new virus is released,

1 users are vulnerable until the new virus has been analyzed and incorporated into the
2 virus databases of the anti-virus applications.

3 Conventional anti-virus software scans e-mail attachments looking for
4 applications that match a virus fingerprint. It can take hours before a fingerprint can be
5 determined, tested, and then made available to anti-virus software customers and if
6 Internet access is adversely impacted by a virus, then it may be extremely difficult to
7 download the anti-virus software update.

8
9 BRIEF SUMMARY OF THE INVENTION

10 Thus, there is a need for a method and system for quickly and preemptively
11 containing and controlling the outbreak of destructive software applications sent in an
12 electronic messaging system. There is also a need for software that can contain the
13 spread of a virus until anti-virus software can be updated to handle a new virus
14 infestation.

15 Accordingly, the preferred embodiments of the present invention provide a
16 system and method for quickly and preemptively controlling the outbreak of destructive
17 software applications sent in an electronic messaging system.

18 The preferred embodiments of the present invention also provide a system and
19 method for containing the spread of viruses until virus software can be updated to handle
20 a new virus infestation.

21 The preferred embodiments of the present invention further provide a system and
22 method for detecting electronic viruses and worms in attachments of electronic messages
23 in an electronic messaging system, wherein the system and method include an add-in to
24 the electronic messaging system to monitor incoming or outgoing electronic messages.

25 The preferred embodiments of the present invention also provide a system and
26 method for an add-in to an electronic messaging system to detect electronic viruses and

1 worms in electronic messages, wherein the add-in can be configured to suit the
2 electronic messaging system and reconfigured to accommodate future virus threats.

3 4 BRIEF DESCRIPTION OF THE DRAWINGS

5 The preferred embodiments are illustrated by way of example and not limitation
6 in the following figures, in which:

7 FIG. 1 depicts a schematic diagram of an electronic messaging system according
8 to one embodiment of the present invention;

9 FIG. 2 illustrates an administrative graphical user interface for creating and
10 maintaining the configuration of a system according to one embodiment of the present
11 invention; and

12 FIG. 3 is a flowchart describing a method for blocking identified files according
13 to an embodiment of the present invention.

14 15 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS OF THE 16 INVENTION

17 The present invention solves the problems associated with prior art anti-virus
18 software in containing the spread of software viruses propagated via an electronic mail
19 system. The invention can be implemented as a software application as described below.
20 In a preferred embodiment described below, the present invention is implemented as an
21 add-in to Microsoft Outlook. One of ordinary skill in the art will appreciate that there
22 are many additional embodiments that can be implemented in any electronic message
23 client software package other than Microsoft Outlook using conventional software
24 development techniques from the description that follows.

25 FIG. 1 depicts a network diagram according to one embodiment of the present
26 invention. The diagram shows ISP Network 105 and Intranet 104 connected to the
27 Internet 101 through Router 110 and Router 111. ISP Network 105 shows a network

1 configuration of an Internet Service Provider (ISP). Users dial up to the ISP Network
2 105 to send and receive e-mail through Mail Server 102 and to access the Internet 101.

3 FIG. 1 also shows Intranet 104, a corporate intranet providing users access to
4 local servers, such as mail server 103, as well as Internet 101 access. For example,
5 Corporate User 108 and Corporate User 109 can use an electronic message client
6 software such as Microsoft Outlook to send and receive e-mail through the Microsoft
7 Exchange application running on Mail Server 103. This system also permits users to
8 send and receive Internet e-mail. For example, Corporate User 108 can send an e-mail
9 message to ISP User 106 and ISP User 107. The e-mail message is first transmitted to
10 Mail Server 103. This server determines the messages destination and transmits the
11 message to Mail Server 102. When ISP User 106 or ISP User 107 logs onto ISP
12 Network 105, the users can access Mail Server 102 to retrieve the message sent from
13 Corporate User 108. Although FIG. 1 depicts an Intranet 104 and an ISP Network 105
14 connected to the Internet 101, one of ordinary skill in the art will appreciate that two ISP
15 networks, two Intranets, or multiple ISP networks and Intranets may be connected to the
16 Internet 101 in the same fashion.

17 E-mail messages may simply contain text; however, they may also contain
18 applications included as attachments. Using a conventional e-mail client software such
19 as Microsoft Outlook, a user can easily execute an attached program by clicking on an
20 icon associated with the attached program. If the attached program contains a virus,
21 such as a worm or other malicious or destructive program, the user's computer can
22 become infected and spread the virus or worm to other users.

23 A preferred embodiment of the present invention is implemented as a software
24 application add-in to Microsoft Outlook/Exchange using programming source codes,
25 which can be compiled using a conventional compiler, to generate an executable
26 program module. According to the preferred embodiment, the add-in includes an
27 implementation of a Microsoft Exchange client extension written in the C++

1 programming language and interfacing with Exchange Server through Microsoft's MAPI
2 (Mail API) 1.0 API (Application Program Interface). The add-in includes a dynamic
3 link library (DLL) application that performs file blocking of attached files that are
4 suspected of carrying viruses or worms. Like many other DLLs, the add-in is written so
5 that its routines are shared by more than one application at the same time. It provides
6 protection against e-mail borne worms and viruses by removing the capability to open
7 messages that may contain destructive programs or scripts. While the add-in does not
8 remove the virus, it does render the virus harmless and prevents it from spreading within
9 an Exchange Server Organization.

10 In the preferred embodiment of the present invention, the add-in DLL application
11 is generated for Microsoft Outlook/Exchange. However, one of ordinary skill in the art
12 will appreciate that the type of add-in depends on the type of electronic message client
13 software being used. Thus, any executable program module that is functionally
14 equivalent to a DLL may be generated according to the present invention to perform the
15 same functions done by the add-in DLL application described herein.

16 Furthermore, one of ordinary skill in the art will appreciate that the source codes
17 used to generate the add-in DLL application may be written and compiled in any known
18 or future computer language format, other than C++, while retaining the same
19 programming logic to generate the add-in DLL application in a Microsoft
20 Outlook/Exchange environment or a compatible executable program module in a
21 corresponding electronic message client software.

22 The add-in DLL application protects users of Outlook from newly introduced
23 viruses. It may be used in conjunction with an anti-virus product and protects users of
24 Outlook from a new virus before an update to the anti-virus product is available,
25 downloaded, and installed. For instance, when a user executes an anti-virus software on
26 his or her machine, the anti-virus software may call on the add-in DLL application to
27 perform its file blocking functions.

1 According to an embodiment of the present invention, the behavior of the add-in
2 is configurable. For example, if a new virus appears in the form of a destructive
3 VBScript program named 'love-letter-for-you.txt.vbs', the add-in can be configured to
4 stop users of Outlook from opening any message that contains an attachment with that
5 name. The configuration is centralized at a host Exchange Server and once set it
6 determines the policy for all users of Outlook within an entire Exchange Server
7 Organization.

8 The add-in behavior is determined by its configuration that may be stored in a
9 hidden message contained within a globally replicated public folder. If the user of
10 Outlook is a member of an Exchange Server distribution list named, for example,
11 'Security Administrators', then a menu item named, for example, 'Security
12 Configuration...' is added to that user's Outlook menu. If this menu is selected, a screen
13 such as that shown in FIG. 2 is displayed. The configuration changes made using this
14 dialog are stored in the hidden message contained within the public folder. Prior to
15 saving any changes to the configuration, the administrator may be warned that the
16 configuration changes will effect the entire Exchange Server Organization and prompted
17 to confirm such changes. One of ordinary skill in the art will appreciate that if an
18 electronic message client software other than Microsoft Outlook is used, the add-in
19 configuration may be stored in a hidden message contained within a globally replicated
20 environment that is functionally equivalent to the public folder of Outlook.

21 According to an embodiment of the present invention, if the add-in is not enabled
22 in box 210, the system does nothing. If the add-in is enabled by marking box 210, as
23 shown in FIG. 2, it will not allow users of Outlook to open messages that contain
24 attachments with names matching those listed in the 'Known Virus Filenames' list 215.
25 If 'Restrict Attachment Types' is checked by marking box 220, it will not allow users of
26 Outlook to open or insert an attachment unless its name ends with one of the filename
27 extensions listed in 'Allow Files With Names Ending In' list 225. Alternatively, the list

1 225 may be a 'Restrict Files with Names Ending In' list, which contains file name
2 extensions that may carry viruses or worms. Thus, a check in box 220 will restrict users
3 of Outlook from opening files with such file name extensions.

4 Thus, the add-in configuration can block file types based on the extension, or
5 trailing characters of the filename. For example, in one embodiment of the present
6 invention, the configuration lists the allowed attachment types. One configuration
7 permits users to only open files ending in, for instance, ".asc", ".csv", ".dat", ".doc", or
8 ".gif" listed in 'Allow Files With Names Ending In'. No other attachments can be
9 opened by a user. In another embodiment of the present invention, users are prohibited
10 from opening files ending in specified extensions listed in 'Restrict Files With Names
11 Ending In'. For example, users may be prohibited from opening executable files or files
12 ending in ".vbx", ".exe", ".com", and ".vbs".

13 The value for 'Seconds to cache this configuration on client' determines how
14 often the add-in will open up the hidden message in the public folder to refresh its
15 configuration. A value of 3600 in box 230 means that the add-in refreshes its
16 configuration at most once per hour. This caching insures that the add-in can be used in
17 large Exchange Server installations.

18 One embodiment of the present invention operates according to the flowchart
19 shown in FIG. 3. A user, using a copy of Outlook containing an add-in according to the
20 present invention, opens an e-mail message as shown in box 301. The add-in is
21 activated and reads the local configuration cache 302 in box 303. The local cache
22 includes the time that the cache was last updated and the number of seconds to cache the
23 present configuration.

24 If the number of seconds since the configuration was last updated exceeds the
25 number of seconds to cache the configuration, or if no configuration is presently cached,
26 then the local configuration is expired in box 304. The system then updates the local
27 configuration in box 305.

1 Next, the system reads the list of attachments in box 306 that are included in the
2 e-mail that the user is trying to open. The filenames of the attachments are then
3 compared to the configuration that is presently cached. The system checks to see if any
4 of the attachment filenames match the "Known Virus Filenames" list in box 307. If one
5 or more matches, then the attachments are blocked; an error message is displayed in box
6 309, and the user is prohibited from opening the e-mail message in box 310.

7 Alternatively, the user is allowed to open the e-mail message but not any of its blocked
8 attachments. According to an embodiment of the present invention, the displayed error
9 message of box 309 may indicate that the user is not allowed to open the attachments
10 and/or the Exchange e-mail system is in virus protection mode. Then the user may be
11 prompted to contact the help desk for any questions. If the e-mail message does not
12 contain a blocked file, then the user is allowed to open the e-mail in box 308.

13 The preferred embodiments described above in conjunction with the
14 accompanying figures and source code are intended to be illustrative of how to practice
15 the present invention. The present invention is not limited to these embodiments. One
16 of ordinary skill in the art will understand that the present invention is applicable to any
17 electronic messaging system where viruses and/or other malicious or destructive
18 software can be transmitted. For example, viruses and/or malicious applications can be
19 transmitted across a network to Personal Digital Assistants (PDAs), cell phones, pagers,
20 or any other communication device. The present invention can also be used in any
21 network system including, but not limited to, wireless, satellite, fiber optic, and cable
22 networks.

23 Although only a few exemplary embodiments of this invention have been
24 described in detail above, those skilled in the art will readily appreciate that many
25 modifications are possible in the exemplary embodiments without materially departing
26 from the novel teachings and advantages of this invention. Accordingly, all such
27 modifications are intended to be included within the scope of this invention as defined in

1 the following claims. Furthermore, any means-plus-function clauses in the claims
2 (invoked only if expressly recited) are intended to cover the structures described herein
3 as performing the recited function and all equivalents thereto, including, but not limited
4 to, structural equivalents, equivalent structures, and other equivalents.

5